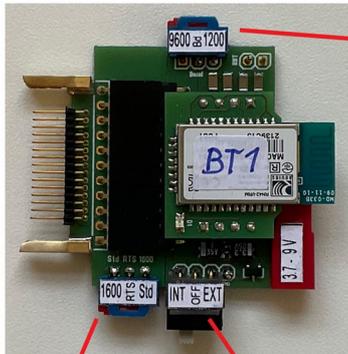


Generic Bluetooth Module for SHARP Pocket Computers – v2.0

(c) 2022 by Spellbound – Extract from full Manual



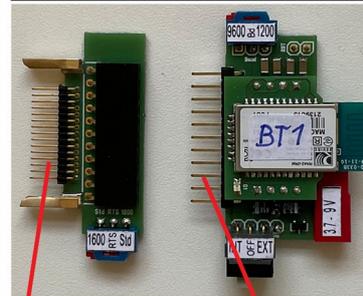
Baudrate switch:
1200 vs 9600

- * Wireless data transmission
- * Suitable for SHARP 15-pin and 11-pin SIO

Hardware handshake switch: PC-1600 style vs 13xx/E500 style

Power supply switch: Internal (Pocket) vs. External (3.7V-9V battery)

Modular Connector Design



15pin - e.g. PC-1600, PC-1350/60, PC-E500 families

11pin - e.g. PC-G850, PC-E220 families

To replace all the different cables, for PC or MAC connection with one reliable, full speed Bluetooth-module with bidirectional UART hardware handshake. No XON/XOFF needed.

Capable of 8-bit binary program/data transfer.

By the configuration switches the module can easily be adopted to any type of SHARP Pocket that is natively equipped with an RS-232 interface (15-pin or 11-pin). No AT-commands needed for config.

CE-133T or equivalent can be modded as a housing with 15-pins.



One fits all

PC-1600



PC-1350/60 and PC-1450/75 families with external power supply



PC-E500/S family



PC-G850V/S and PC-E220 families



Table of Contents

Introduction.....	3
Structure.....	3
Hardware Functions.....	4
BT-Pairing and virtual COM-Ports.....	5
BT-Connection.....	6
Serial Settings – Hardware and Software.....	7
Settings Table.....	8
Program and Data Transfer.....	9
Kit Assembly.....	10
Appendix A: CE-133T Mod as a Shell.....	13
Appendix B: EOL and EOF Configuration.....	14
Appendix C: Symptoms, Causes and Solutions.....	16
Appendix D: Advanced Bluetooth Configuration.....	17
Appendix E: Restore Factory Defaults.....	20

Introduction

This module is a wireless replacement for serial RS-232 cables and level converters, like the CE-160xL, CE-13xT, CE-T800/1 or custom FTDI-chip based USB-to-UART cables.

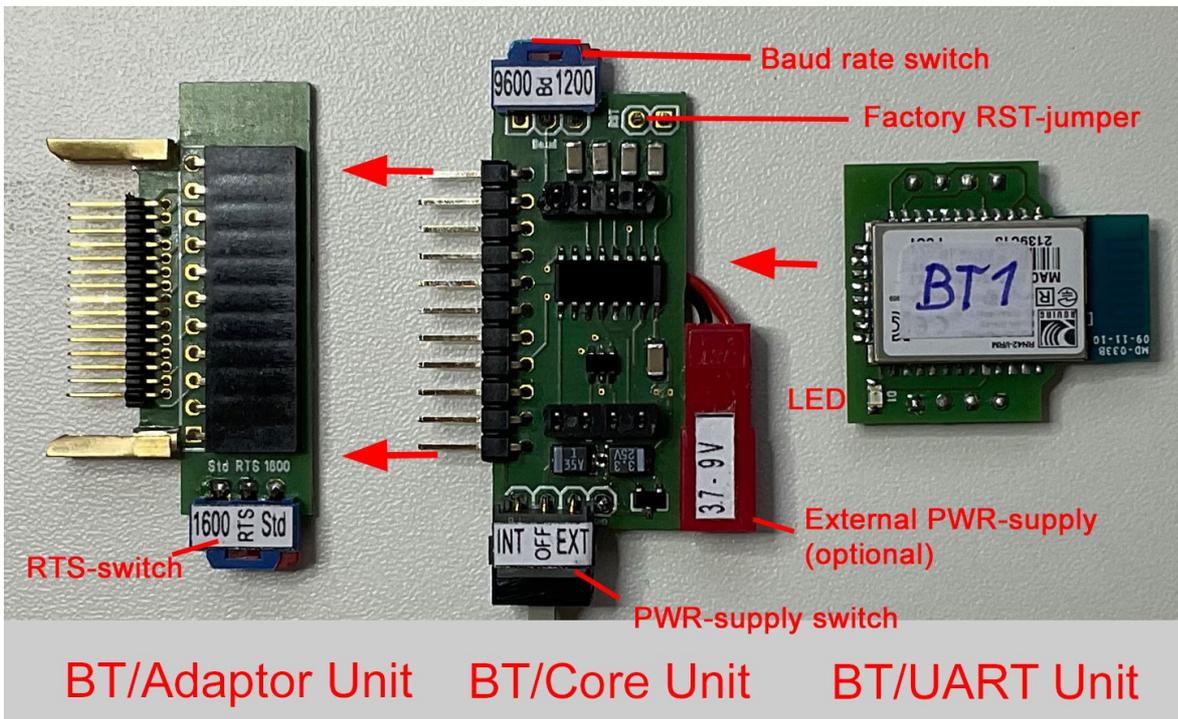
There are several advantages of the module over cable based solutions:

- Wireless communication to a PC/MAC (obvious)
- Reliable, full speed data transfer with bidirectional hardware handshake:
The most common cables (e.g. CE-133T) do not support RS-232 hardware handshake at all, so the XON/XOFF protocol (software handshake) and typically a slower baud rate than the pockets maximum must be used instead.
However, if you have a cable that supports bidirectional RTS/CTS hardware handshake for the PC-1600 it will probably not support hardware handshake for the PC-E500(S) and PC-1350/60, and vice versa, because of incompatible signal interpretations (RTS vs. RTR) of these models - although they use the same physical 15-pin plug.
This BT-module is able to compensate that via a switch.
- 8-bit binary data/program transfer capability
If your pocket computer supports serial, binary data transfer, like the PC-1600 (e.g. by BSAVE/BLOAD) – the BT-module can handle that as well.
- One-fits-all:
The BT-module is designed to support SHARP pockets with a 15-pin serial port as well as those with a 11-pin serial port.

Structure

The BT-module consists of three sub-modules or units:

- BT/UART Unit
Bluetooth chip with a custom 8-pin shield.
- BT/Core Unit
Custom level-converter and adaptor for the SHARP 11-pin serial interface
- BT/Adaptor Unit
Custom adaptor from SHARP 11-pin to SHARP 15-pin serial interface including a switch for the hardware handshake style (i.e. PC-1600 vs. other 15-pin)



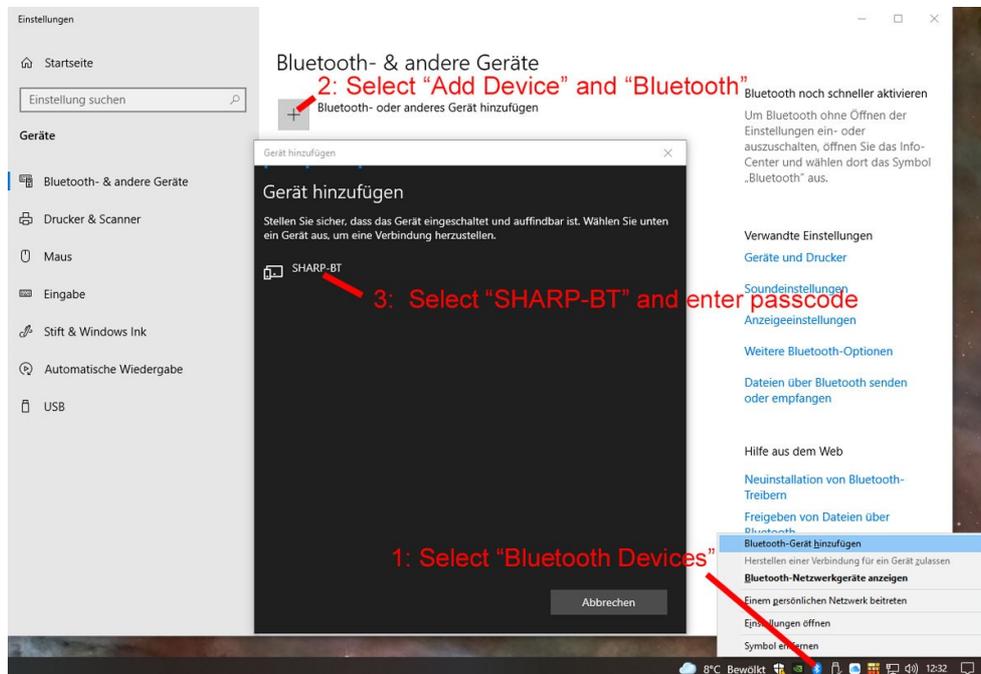
Hardware Functions

The picture above is also showing the switches, connectors, LED and jumper positions:

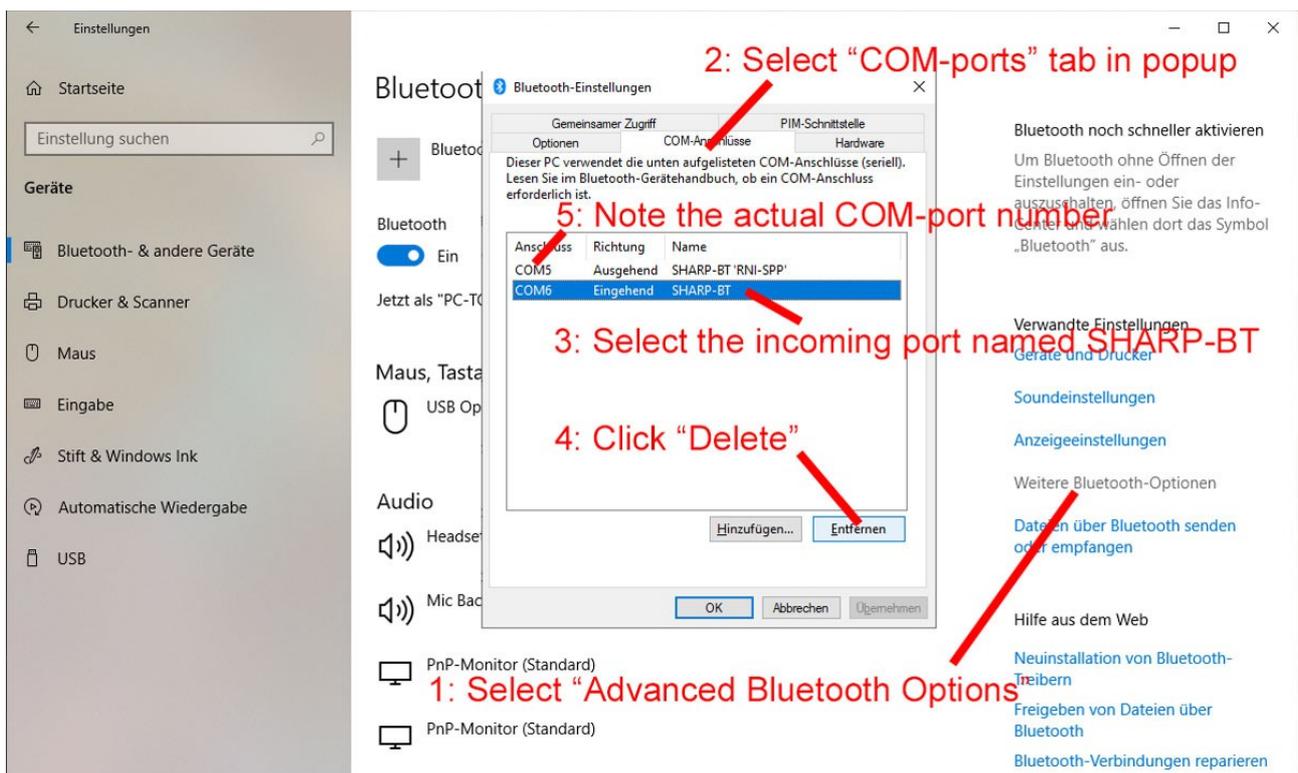
- **RTS-switch**
Toggles the hardware handshake pinout for the 15-pin interface between PC-1600 style and standard style. Every SHARP pocket computer with a 15-pin interface except the PC-1600 needs the standard mode. Technically standard mode pockets are those that provide an RTR signal at pin-11 of the 15-pin serial interface. In contrast the PC-1600 interface only exposes the RTS signal at pin-4 and none at pin-11, but that too has RTR-semantics.
- **Baud rate switch**
Selects between 9600baud (e.g. PC-1600, PC-E500 series, PC-G8xx series) and 1200baud (e.g. PC-1450/75, PC-13xx series). The baud rate setting has to be identical on the pocket computer, the bluetooth module and the terminal application on your PC/MAC.
- **PWR-supply switch**
Toggles the power supply source for the BT-module between internal (i.e. batteries of the pocket) vs. off vs. external (i.e. power supply connected to the External PWR-supply). The off state is useful for pockets, that have active power supply at their interface even when switched off, like the PC-E500 and the PC-G850V/S.
- **External PWR-supply**
An external power supply of 3.7V (e.g. a lithium/ion battery) up to 9V can be connected to unburden the batteries of the pocket. The kit option does not contain the actual plug, since it depends on your actual power supply what kind of plug makes sense, if any. 3 or 4 AA-batteries in series is optimal. When the BT-module is used with pockets that are driven by coin cells, like the PC-14xx and PC-13xx, the external power supply is mandatory.
ATTENTION:
Take care of correct polarity (+/-), otherwise the module may be destroyed.
- **Factory RST-jumper**
This jumper can be used to force a reset to the factory default settings of the Bluetooth chip, so it is not just a session/state reset and should only be used in emergency cases.
- **LED**
The LED on the BT/UART unit has three states when the module is powered:
Flashing about 1 times per sec means "Ready to pair or connect".
Flashing about 10 times per sec means "In command mode".
Off means connected and ready for data transfer.

BT-Pairing and virtual COM-Ports

The Bluetooth pairing process is just like with any other Bluetooth v2 device. When powered the BT-module's LED should flash 1x per sec. Then it is visible to your PC/Laptop/MAC and the pairing can be initiated from those peers. The name of the module is "SHARP-BT" and the passcode is 1234. For Windows 10 the dialog looks like this:



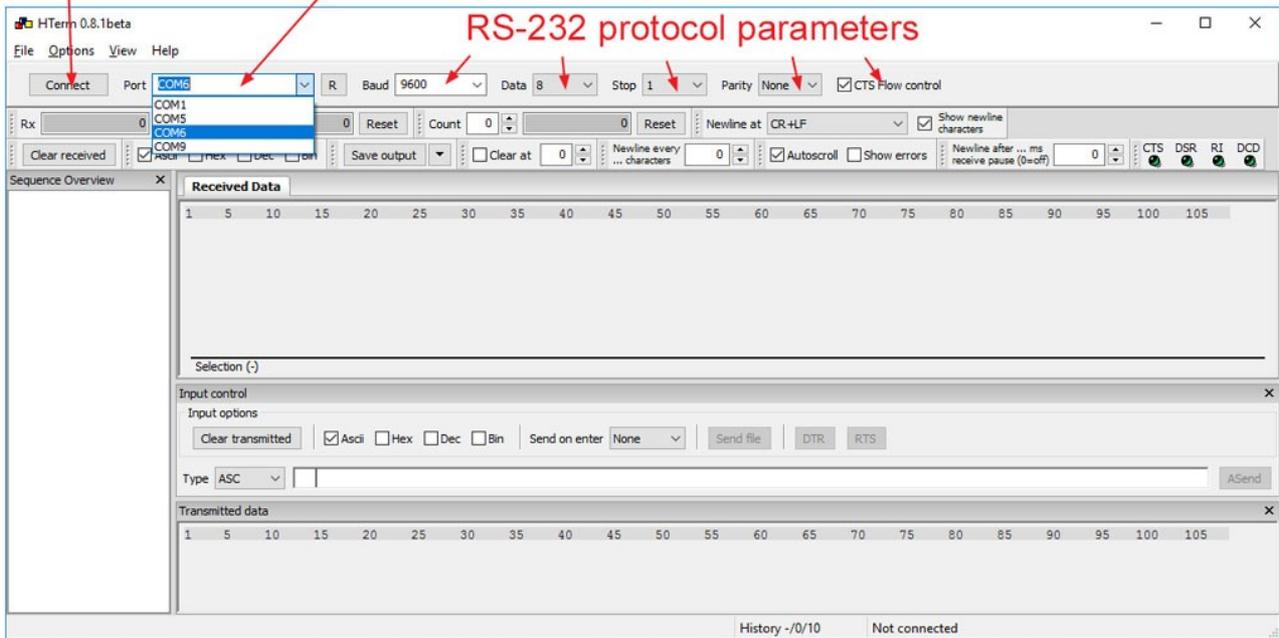
Now on Windows you have to determine the virtual COM-port that has been created for the module. You will need this in the terminal application in order to connect to the module. And you should also get rid of the superfluous incoming COM-port Windows has created in order to keep your system clean:



BT-Connection

As a result of the BT-pairing process the host operating system (Windows, OSX, Linux) creates at least one virtual COM-port. The terminal apps typically enumerate all existing COM-ports. In OSX the ports are named, in Windows they only have numbers, so you need to look up the right one (see above). You always need the outgoing port for the device *SHARP-BT*. In the following sections all settings are illustrated via the terminal app *HTerm* on Windows 10 (recommended). On OSX you can use e.g. *CoolTerm*. Here are the basics:

Connect button Virtual COM port enumeration



A successful connection typically is indicated by the terminal app by showing some active signals like CTS, DSR or DCD:



You should always disconnect in the terminal app before you turn the modules power off, to avoid inconsistent states between the module and the terminal app. However this is not critical. In the worst case you have to restart the terminal app.

ATTENTION

When the BT-module is powered but not connected it performs an active scan for BT-peers, which is the most power consuming mode of the module. So you shouldn't leave the module powered but unconnected for a longer period of time to safe battery life.

Serial Settings – Hardware and Software

The fundamental principle of RS-232 communication in general is the alignment of the protocol parameters. They have to be identical for the pocket computer and the terminal program used at the host. Since the BT-module can be thought of as an 'intelligent' virtual RS-232 cable, its settings have to be identical too, in order to establish a proper data tunnel. The essential RS-232 protocol parameters of the BT-module are:

1. Baud rate: 9600 or 1200 – selectable by the Baud rate switch.
2. Data bits: 8
3. Parity bit: None
4. Stop bits: 1
5. RTS/CTS hardware handshake: On
6. DTR/DSR hardware handshake: Not supported / Off
7. XON/XOFF software handshake: Not supported / Off
(i.e. the BT-module does not generate or interpret XON- and XOFF-chars but treats them as normal characters and just transfers them)

ATTENTION

Since these parameter settings are fixed for the BT-module (except the baud rate), the pocket computers serial settings as well as those of the terminal app must follow this specification. Furthermore the BT-modules RTS switch must be set to the correct position depending on the type of pocket computer the module is connected to. If one of these settings is inconsistent you will experience errors or problems when transferring programs or data but not at connection time. A configuration mismatch is the most frequent error source in RS232 communication in general.

The following table might be the most important part in this manual. It shows the correct setting for each type of supported SHARP pocket computer. Please follow this guide carefully, especially when you switch between different types of SHARP pocket computers.

Settings Table

SHARP PC	Pocket Computer RS-232 Setting Sequence (BASIC or Menu)	BT-Module Baude Rate Switch	BT-Module RTS Switch	BT-Module External Power Supply	Terminal App Baud Rate Setting
13xx 14xx	OPEN "1200,N,8,1,A, L , &1A " CLOSE	1200	Std	Mandatory	1200
2500	OPEN "1200,N,8,1,A, L , &1A " CLOSE	1200	Std	Optional	1200
1600	SETCOM "COM1:",9600,8,N,1,N,N SNDSTAT "COM1:",59 RCVSTAT "COM1:",61 OUTSTAT "COM1:"	9600	1600	Optional	9600
E220	Menu->Sio->Format: baud rate = 1200 data bit = 8 stop bit = 1	1200	N.A.	Optional	1200
E500	OPEN "9600,N,8,1,A, L , &H1A ,N,N" CLOSE	9600	Std	Optional	9600
G8xx	Menu TEXT->Sio->Format: baud rate = 9600 data bit = 8 stop bit = 1 flow = RS/CS	9600	N.A.	Optional	9600

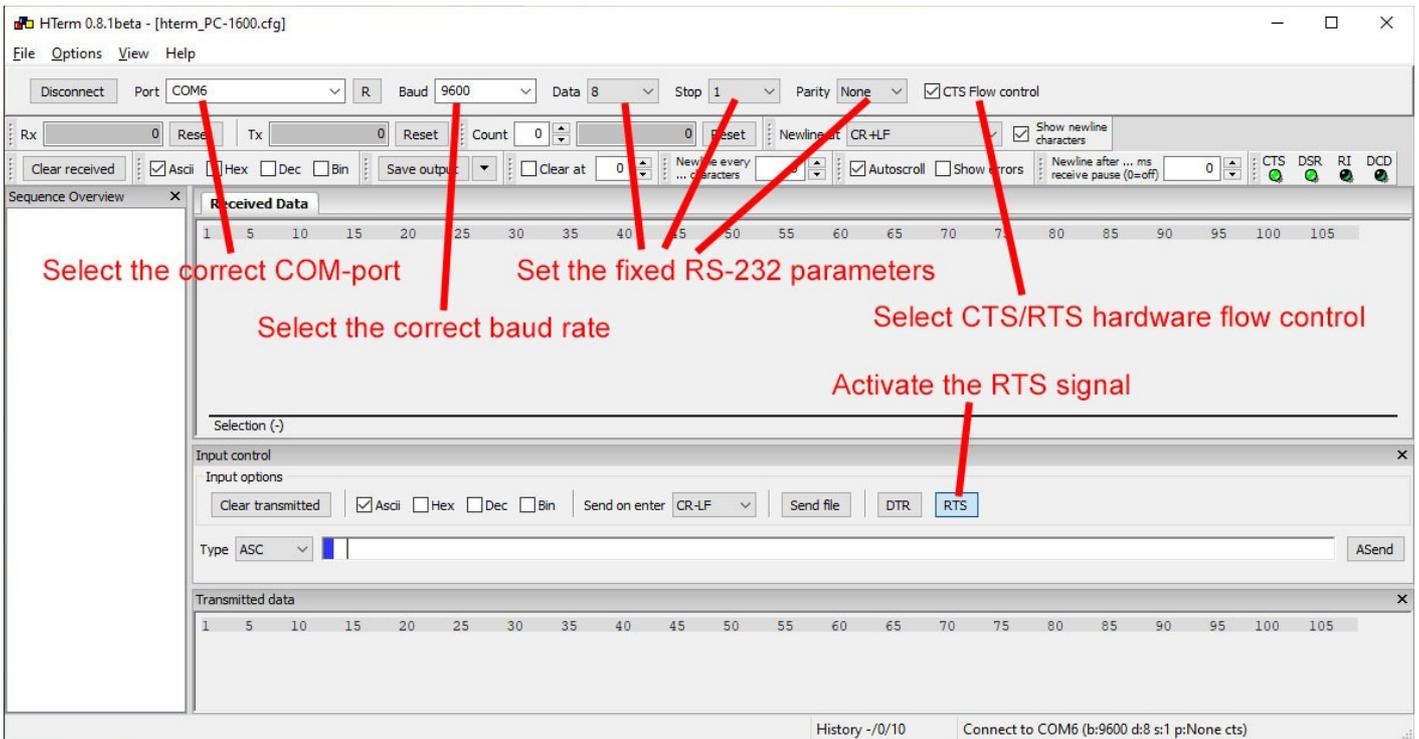
RS-232 settings made on the pocket computer are persistent until a hard reset is performed or the batteries are changed. So they survive a normal power-off/power-on cycle. In other words these are 'one-off' settings.

The End-Of-Line (EOL) settings (red) and the End-Of-File (EOF) settings (blue) may be changed according to the desired file type:

- C = CR only
- F = LF only
- L = CR + LF

All other RS-232 settings - data bits, parity, stop bits, and especially the RTS/CTS handshake - must be set in the terminal app exactly as stated above.

Also the RTS signal should be activated in order to allow the BT-module to send data. Otherwise commands like SAVE may be blocked (waiting indefinitely long).



Program and Data Transfer

Once a connection is established and all settings (hardware and software) are done correctly as described above, the BT-module acts like a virtual RS-232 cable and you can e.g. SAVE and LOAD programs or transfer data via PRINT# or INPUT#. Please see the manual of your pocket computer for the details. Here is an overview for the basics:

SHARP PC	Saving program to host	Loading program from host
13xx	OPEN	OPEN
14xx	SAVE	LOAD
2500	CLOSE	CLOSE
1600	SAVE"COM1 : ", A	LOAD"COM1 : "
E220	Menu->Sio->Save or in MON mode: W-command	Menu->Sio->Save or in MON mode: R-command
E500	SAVE	LOAD
G8xx	Menu TEXT->Sio->Save or in MON mode: W-command	Menu TEXT->Sio->Load or in MON mode: R-command

ATTENTION

When loading programs (or data), you should first give the respective load command on the pocket computer (which will then wait for data) and then send the file via the terminal app.

In *HTerm* you can send files to the pocket computer via the 'Send file'-button. Received data is shown in the respective text field, and you can save that data via the 'Save output'-button.